This listing of claims will replace all prior versions, and listings, of claims in the application:

The Status of the Claims

1. (Currently Amended) A method of cache intervention comprising:

reading a memory block into a first cache from a main memory;

copying the memory block from the first cache to a second cache in response to a read request associated with the second cache;

tagging the memory block in the first cache as non-modified;

detecting a read request associated with a third cache that hits the first cache and the second cache;

if the memory block in the first cache and the memory block in the second cache are both tagged as non-modified, selecting the first cache or the second cache as an arbitration winner cache based on a ~~predetermined~~ fixed hardware arbitration hierarchy; and

providing a copy of the memory block from the arbitration winner cache responsive to the read request.

2. (Original) A method as defined in claim 1, wherein tagging the memory block in the first cache as non-modified comprises tagging the memory block in the first cache as exclusive.

3. (Original) A method as defined in claim 1, wherein tagging the memory block in the first cache as non-modified comprises tagging the memory block in the first cache as shared.

4. (Previously Amended) A method as defined in claim 1, further comprising tagging the memory block in the second cache as shared-.

5. (Original) A method as defined in claim 1, wherein detecting a read request comprises snooping a bus.

6. (Original) A method as defined in claim 1, wherein detecting a read request comprises accessing a cache directory.

7. (Previously Amended) A method as defined in claim 1, further comprising tagging the memory block in the third cache as non-modified.

8. (Previously Amended) A method of cache intervention comprising:

reading a memory block into a first cache from a main memory;

tagging the memory block in the first cache as non-modified;

detecting a read request associated with the memory block that hits an agent associated with a second cache;

providing a copy of the memory block from the first cache responsive to the read request;

detecting a read request associated with the memory block that hits an agent associated with a third cache;

determining an arbitration winner cache between the first cache and the second cache if the memory block in the first cache is tagged as non-modified and the memory block in the second cache is tagged as non-modified;

copying the memory block from the arbitration winner cache to the third cache; and

tagging the memory block in the third cache as non-modified, wherein determining an arbitration winner cache comprises selecting a directory entry associated with the first cache or the second cache.

9. (Previously Amended) A method of cache intervention comprising:

reading a memory block into a first cache from a main memory;

tagging the memory block in the first cache as non-modified;

detecting a read request associated with the memory block that hits an agent associated with a second cache;

providing a copy of the memory block from the first cache responsive to the read request;

detecting a read request associated with the memory block that hits an agent associated with a third cache;

determining an arbitration winner cache between the first cache and the second cache if the memory block in the first cache is tagged as non-modified and the memory block in the second cache is tagged as non-modified;

copying the memory block from the arbitration winner cache to the third cache; and

tagging the memory block in the third cache as non-modified, wherein determining an arbitration winner cache comprises usage of a back-off based arbitration mechanism.

10. (Previously Amended) A multi-processing computing device comprising:

a first processing agent including a first processor and a first cache;

a second processing agent including a second processor and a second cache, the second processing agent being coupled to the first processing agent via a cache interconnect;

a third processing agent including a third processor and a third cache, the third processing agent being coupled to the first processing agent and the second processing agent via the cache interconnect;

an arbitration circuit enforcing a fixed cache intervention priority between the first processing agent, the second processing agent and the third processing agent; and

a main memory coupled to the first processing agent, the second processing agent and the third processing agent via a main memory interconnect, the first processing agent to (i) read a memory block into the first cache from the main memory via the main memory interconnect, (ii) tag the memory block in the first cache with an exclusive tag, and (iii) supply the memory block tagged exclusive to the second cache via the cache interconnect.

11. (Original) A multi-processing computing device as defined in claim 10, wherein the first processing agent is to detect a read request associated with the memory block by the second processing agent and retag the memory block in the first cache as shared in response to detecting the read request.

12. (Original) A multi-processing computing device as defined in claim 11, wherein the first processing agent is to detect the read request associated with the memory block by snooping the cache interconnect.

13. (Previously Amended) A multi-processing computing device as defined in claim 11, wherein the first processing agent or the second processing agent are to supply the memory block tagged shared to a third cache via the cache interconnect.

14. (Original) A multi-processing computing device comprising:

a first processing agent including a first processor, a first cache, and a signal input, the first cache to store a memory block in a shared state;

a second processing agent including a second processor, a second cache, and a signal output, the second cache to store the memory block in the shared state, the signal output being coupled to the signal input; and

a third processing agent including a third processor and a third cache, the second processing agent to supply the memory block tagged shared to the third cache, the second processing agent to prevent the first processing agent from supplying the memory block tagged shared to the third cache by asserting the signal output.

15. (Original) A multi-processing computing device as defined in claim 14, further comprising a logical OR unit, the logical OR unit including a first OR input, a second OR input, and an OR output, the first OR input being coupled to the signal output of the second processing agent, the second OR input being coupled to the third processing agent, the OR output being coupled to the signal input of the first processing agent.

16. (Original) A multi-processing computing device as defined in claim 14, further comprising a logical OR unit and a fourth processing agent, the logical OR unit including a first OR input, a second OR input, and an OR output, the first OR input being coupled to the signal output of the second processing agent, the second OR input being coupled to the fourth processing agent, the OR output being coupled to the signal input of the first processing agent.

17. (Currently Amended) A computer comprising:

a first microprocessor including a first cache, the first cache to store a first copy of a memory block in a non-modified state;

a second microprocessor including a second cache to store a second copy of the memory block in the non-modified state;

a third microprocessor including a third cache; and

a main memory coupled to the first microprocessor, the second microprocessor and the third microprocessor, the first microprocessor or the second microprocessor to supply the third cache with a third copy of the memory block based on a physically predetermined arbitration hierarchy while the first copy of the memory block and the second copy of the memory block are in the non-modified state.

18. (Previously Amended) A computer as defined in claim 17, wherein the first microprocessor is to supply the second cache with the second copy of the memory block while the first copy of the memory block is in an exclusive state or a shared state.

19. (Previously Amended) A computer as defined in claim 17, wherein the main memory is operatively connected to the first microprocessor and the second microprocessor by a main memory bus, the first microprocessor to directly supply the second cache with the second copy of the memory block while the first copy of the memory block is in the non-modified state.

20. (Currently Amended) A computer ~~as defined in claim 19,~~ ~~further~~ comprising:

a first microprocessor including a first cache, the first cache to store a first copy of a memory block in a non-modified state;

a second microprocessor including a second cache to store a second copy of the memory block in the non-modified state;

a third microprocessor including a third cache; and

a main memory coupled to the first microprocessor, the second microprocessor and the third microprocessor, the first microprocessor or the second microprocessor to supply the third cache with a third copy of the memory block based on a predetermined arbitration hierarchy while the first copy of the memory block and the second copy of the memory block are in the non-modified state, wherein the main memory is operatively connected to the first microprocessor and the second microprocessor by a main memory bus, the first microprocessor to directly supply the second cache with the second copy of the memory block while the first copy of the memory block is in the non-modified state;

a mother board coupled to the first microprocessor;

a hard drive coupled to the first microprocessor; and

a graphics card coupled to the first microprocessor.

21. (Original) A computer as defined in claim 20, further comprising:

an input device coupled to the first microprocessor; and

an output device coupled to the first microprocessor.

22. (Previously Amended) A computer as defined in claim 21, wherein the input device comprises at least one of a keyboard, a mouse, a track pad, an isopoint, a microphone, or a graphics tablet.

23. (Previously Amended) A computer as defined in claim 21, wherein the output device comprises at least one of a display, a printer, a modem, a network card, or a speaker.

24. (Currently Amended) A method of cache intervention comprising:

storing a memory block in a first cache in a shared state;

storing the memory block in a second cache in the shared state;

detecting a read request associated with the memory block by an agent associated with a third cache while the memory block in the first cache is in the shared state and the memory block in the second cache is in the shared state;

selecting the first cache or the second cache to provide a copy of the memory block to the third cache based on a ~~predetermined~~ permanent hierarchy between the first and second caches; and

copying the memory block to the third cache in accordance

with the selection.

25. (Original) A method as defined in claim 24, further comprising preventing a third cache from copying the memory block to the second cache.

26. (Previously Amended) A method of cache intervention comprising:

storing a memory block in a first cache in a shared state;

detecting a read request associated with the memory block by an agent associated with a second cache while the memory block in the first cache is in the shared state;

copying the memory block from the first cache to the second cache in response to detecting the read request while the memory block in the first cache is in the shared state; and

preventing a third cache from copying the memory block to the second cache, wherein preventing a third cache from copying the memory block to the second cache comprises asserting a "back-off" signal.

27. (Previously Amended) A method of cache intervention comprising:

storing a memory block in a first cache in a shared state;

detecting a read request associated with the memory block by an agent associated with a second cache while the memory block in the first cache is in the shared state;

copying the memory block from the first cache to the second cache in response to detecting the read request while the memory block in the first cache is in the shared state; and

preventing a third cache from copying the memory block to the second cache, wherein preventing a third cache from copying the memory block to the second cache comprises determining an arbitration winner cache represented in a cache directory.

28. (Currently Amended) A method of cache intervention comprising:

storing a first copy of a memory block in a first cache in an exclusive state or a shared state;

detecting a read request corresponding to a second cache and associated with the memory block;

supplying the second cache with a second copy of the memory block while the first copy of the memory block is in the exclusive state or the shared state without accessing a main memory; and

preventing a third cache having a third copy of the memory block from supplying the second copy of the memory block to the second cache if the first and third copies of the memory block are in the shared state and the first cache has a higher cache intervention priority under a ~~predetermined~~ fixed hierarchy defined between the first and second caches.

29. (Original) A method as defined in claim 28, wherein detecting a read request associated with the memory block comprises snooping a bus.

30. (Original) A method as defined in claim 28, wherein detecting a read request associated with the memory block comprises using a cache directory.

31. (Previously Amended) A method as defined in claim 28, wherein supplying the second cache with a second copy of the memory block while the first copy of the memory block is in the exclusive state or the shared state comprises supplying the second cache with the second copy of the memory block while the first copy of the memory block is in the exclusive state.

32. (Previously Amended) A method as defined in claim 28, wherein supplying the second cache with a second copy of the memory block while the first copy of the memory block is in the exclusive state or the shared state comprises supplying the second cache with the second copy of the memory block while the first copy of the memory block is in the shared state.